The Astrophysical Journal, 935:87 (8pp), 2022 August 20 https://doi.org/10.3847/1538-4357/ac7fa2

Foraging with MUSHROOMS: A Mixed-integer Linear Programming Scheduler for Multimessenger Target of Opportunity Searches with the Zwicky Transient Facility

B. Parazin[1,2], Michael W. Coughlin[2], Leo P. Singer[3,4], Vaidehi Gupta[5], and Shreya Anand[6]
[1] College of Science, Northeastern University, Boston, MA 02115, USA; Parazin.b@northeastern.edu
[2] School of Physics and Astronomy, University of Minnesota, Minneapolis, MN 55455, USA
[3] Astrophysics Science Division, NASA Goddard Space Flight Center, Code 661, Greenbelt, MD 20771, USA
[4] Joint Space-Science Institute, University of Maryland, College Park, MD 20742, USA
[5] Indian Institute of Technology, Kharagpur, West Bengal, 721302, India
[6] Division of Physics, Mathematics and Astronomy, California Institute of Technology, Pasadena, CA 91125, USA
Received 2022 March 22; revised 2022 June 18; accepted 2022 July 7; published 2022 August 18

Abstract

Electromagnetic follow-up of gravitational-wave detections is very resource intensive, taking up hours of limited observation time on dozens of telescopes. Creating more efficient schedules for follow-up will lead to a commensurate increase in counterpart location efficiency without using more telescope time. Widely used in operations research and telescope scheduling, mixed-integer linear programming is a strong candidate to produce these higher-efficiency schedules, as it can make use of powerful commercial solvers that find globally optimal solutions to provided problems. We detail a new target-of-opportunity scheduling algorithm designed with Zwicky Transient Facility in mind that uses mixed-integer linear programming. We compare its performance to `gwemopt`, the tuned heuristic scheduler used by the Zwicky Transient Facility and other facilities during the third LIGO–Virgo gravitational-wave observing run. This new algorithm uses variable-length observing blocks to enforce cadence requirements and to ensure field observability, along with having a secondary optimization step to minimize slew time. We show that by employing a hybrid method utilizing both this scheduler and `gwemopt`, the previous scheduler used, in concert, we can achieve an average improvement in detection efficiency of 3%–11% over `gwemopt` alone for a simulated binary neutron star merger data set consistent with LIGO–Virgo's third observing run, highlighting the potential of mixed-integer target of opportunity schedulers for future multimessenger follow-up surveys.

Unified Astronomy Thesaurus concepts: Gravitational wave astronomy (675); Astronomical methods (1043); Gravitational waves (678); Transient detection (1957); Transient sources (1851); Time domain astronomy (2109)

1. Introduction

The detection of GW170817 (Abbott et al. 2017a) in 2017 August signaled the beginning of a new era of multimessenger astronomy, promising advances in *r*-process nucleosynthesis (e.g., Chornock et al. 2017; Coulter et al. 2017; Cowperthwaite et al. 2017; Pian et al. 2017), the neutron star equation of state (e.g., Abbott et al. 2018; Radice et al. 2018; Coughlin et al. 2019, 2018, 2020; Dietrich et al. 2020a), and the value of the Hubble constant (e.g., Abbott et al. 2017b; Hotokezaka et al. 2019; Dietrich et al. 2020a). This was thanks to the detection of GW170817 (Abbott et al. 2017a) and its electromagnetic counterparts: a kilonova (ultraviolet/optical/near-IR emission generated by the radioactive decay of *r*-process elements; e.g., Evans et al. 2017; Kasliwal et al. 2017; Kilpatrick et al. 2017; Pian et al. 2017; Shappee et al. 2017; Smartt et al. 2017), a short gamma-ray burst (e.g., Goldstein et al. 2017), and an afterglow (e.g., Hallinan et al. 2017; Troja et al. 2017). However, since then, no further electromagnetic counterparts to gravitational-wave detections have been confirmed, despite several other detected binary neutron star and neutron star black hole mergers during LIGO–Virgo's third observing run (Abbott et al. 2021). This can mostly be explained by the localization areas of neutron star containing mergers being much larger than expected, (e.g., Coughlin et al. 2020; Petrov et al. 2022), making efficient observation planning all the more important. With, on average, a much larger area than previously thought to search, there are many more choices for potential schedules, but it will take an optimal scheduler to maximize scientific output.

Fundamentally, telescope scheduling software determines which fields to observe in what order, subject to environmental and programmatic constraints. In the case of the follow-up of large sky localizations produced by gravitational-wave (e.g., Coughlin et al. 2019a; Anand et al. 2020) or gamma-ray burst (e.g., Coughlin et al. 2019b; Ahumada et al. 2021) events with wide field-of-view surveys such as the Zwicky Transient Facility (ZTF; Bellm et al. 2019c; Graham et al. 2019; Dekany et al. 2020; Masci et al. 2018), the goal is usually to maximize an objective function, which is typically taken to be the integral of the probability skymap over the combined footprint of all the observations, although other choices are possible (Coughlin et al. 2018). These observations should also be completed in the minimal amount of time, as many different science programs time share on the same telescope, and therefore any time saved can be utilized by other science programs (Bellm et al. 2019b).

While, in principle, this could be done manually, schedules designed this way are labor-intensive and sub-optimal, and it is unclear how the heuristics translate to survey effectiveness. Another common approach is the use of "greedy" algorithms, which compute a metric or score for each possible target, select

the target with the current highest value, observe it, and then repeat the process. This is a ubiquitous approach, adapted by commonly used packages ranging from `Astroplan` (Morris et al. 2018) to `gwemopt` (Coughlin et al. 2018, 2019c; Almualla et al. 2020) for a variety of purposes. Unfortunately, due to the inability to plan ahead, the fields chosen are not optimal; instead, an optimal schedule not only accounts for the current possible observations but also for past and potential future observations to maximize the scientific output from those observations, such as the ZTF's need for a minimum of 30 minutes cadence when searching for transients to rule out asteroids.

Unfortunately, the scheduling problem is NP-complete and the number of observing sequences is combinatorially large. A well-known model that can make these problems computationally tractable is the use of integer linear programming (ILP); ILP problems have variables that take only discrete integer values, linear objective functions, and linear constraints. In the following, we will also use mixed ILP, which can include some nonintegral variables. One popular application of this in astronomy is within the Las Cumbres Observatory (LCO)[7] scheduler, who operate a network of identical imagers and spectrographs. Their scheduler (Lampoudi et al. 2015) uses ILP to maximize the total number of observations obtained, weighted by the priority assigned to them by the Time Allocation Committee (TAC). Atacama Large Millimeter/submillimeter Array solves a similar ILP model to maximize TAC-assigned scientific priorities, program completion, and telescope utilization (Solar et al. 2016). The ZTF's time-allocation scheduler uses ILP to solve both program-level and global scheduling constraints, and to optimally order individual observational blocks (Bellm et al. 2019c); however, the schedulers used to plan within each observation block do not all use ILP, such as the greedy scheduler `gwemopt`. Bellm et al. (2019a), a paper whose authorship spans many surveys and open-source astronomy software producers, advocate for community emphasis on the use of quantitative objective functions and ILP-based scheduling approaches to address the rapid proliferation of instruments, many of which will benefit from coordination.

In this paper, we introduce MILP-using scheduler of sky localization maps (MUSHROOMS), an MILP-based scheduler for multimessenger follow-up with wide field-of-view surveys. We structure the paper as follows. We start by describing requirements faced by the ZTF gravitational-wave follow-up observations in Section 2. We then introduce MUSHROOMS and describe the scheduling algorithm in Section 3, laying out its MILP formalism and the reasoning behind certain design decisions. In Section 4, we use MUSHROOMS to schedule simulated skymaps based on LIGO–Virgo's third observing run and characterize its runtime, efficiency, and other relevant metrics, and summarize our results and future outlook in Section 5.

2. Observing Requirements

Multimessenger astronomy supplements electromagnetic observations with observations using other information carriers such as gravitational waves or neutrinos. Since 2018, the ZTF has been used for target-of-opportunity, multimessenger follow-up searches, both searching for the sources of gravitational-wave detections during the third LIGO–Virgo observing run (Coughlin et al. 2019a; Anand et al. 2020; Kasliwal et al. 2020), and gamma-ray bursts from detectors like the Fermi Gamma-ray Burst Monitor (Coughlin et al. 2019b; Ahumada et al. 2021). However, there are a number of factors one has to consider when designing schedules for such systems, both in terms of general observational requirements for ground-based surveys and certain ZTF-specific restrictions or demands. For example, targets are only observable at night and when they are above a minimum altitude from the horizontal (i.e., below a minimum airmass). There are also common sense constraints: for example, the scheduler cannot schedule more than one field observation at the same time, and it must restrict observations to the window of time available for observing. In addition, there are also limits imposed by the telescope and observing system itself, such as slew speed.

There are also a number of multimessenger transient follow-up restrictions that must be accounted for. For example, for the ZTF gravitational-wave follow-up program, there is a 30 minute cadence requirement, observed once in r and g bands, which serves to both eliminate asteroids and to gain color information about detected transients. This requirement imposes not only a limit on the return time, but also must account for the filter exchange time within the ZTF, which is 2 minutes long. Another special feature of the ZTF follow-up is that the system uses a fixed grid of reference images, a preset selection of a limited number of telescope pointings to choose from.

In addition to the requirements, the goal is to limit the total amount of time required for these observations through the selection of an objective function, whose choice we will describe below.

3. Scheduling Algorithm and MILP Formulation

Due to the design of the ZTF survey and data system, the ZTF has a fixed grid of 1778 telescope pointings. Given a probability density map in R.A. and decl., a span of time of t_0 to $t_o + T$, and a fixed-exposure time Δt, the goal is to produce a schedule that meets the observing requirements laid out in Section 2 by selecting a set of fields S to observe and arranging them in time (with repeats). The objective is to maximize the total probability density contained in the area observed by at least one field in S minus a penalty factor proportional to the amount of fields observed with the proportionality constant p. We maximize the probability density contained because the overall goal of this scheduler is to identify new transients that could potentially be a gravitational-wave source for a follow-up observation. This is done by comparing observations to reference images to find new sources, so maximizing the probability density observed in theory maximizes the probability of detecting the source for follow-up. We introduced the penalty factor p with the other survey priorities in mind. It allows the user to restrict the search to only fields that introduce more than p to the total probability observed. The expected range of p is [0, 0.02], though it has to be manually selected for each skymap. While not making it impractical for use in scheduling, using it does require some extra effort from the user to determine the trade-off between observing time and detection probability best for their situation. This creates shorter duration schedules that only target the highest-probability fields and are less intrusive to the other programs;

[7] https://lco.global/

with a value of zero for p, the schedule will fill all available time.

MUSHROOMS (Parazin 2022) is a python-based mixed-integer scheduler that uses the commercially available software `Gurobi`, which is free with an academic license. When the network of ground-based gravitational-wave detectors localize a new event, they release a probability map of where in the sky the source is most likely located (Singer & Price 2016), which MUSHROOMS takes as one of its inputs. An example schedule overlaid on its corresponding probability map is shown in Figure 2. For each given source-localization probability map, referred to as a skymap from here on, the MUSHROOMS algorithm works in a three-step process: a preliminary pruning step, a block-division step, and an observation sequencing step. A flowchart illustrating the whole algorithm can be seen in Figure 1.

In the pruning step, MUSHROOMS reduces the field grid to a user-provided number of fields using a max-weighted coverage algorithm (Nemhauser et al. 1978). This is done to reduce the runtime during the block-division step.

In the next step, the block-division step, a number of observing blocks are constructed out of the field shortlist produced by the pruning step. MUSHROOMS defines an observing block by a start time, an expected end time, and a collection of fields that are visible for the entire expected duration. To observe each block, all the fields within it are observed in the same filter, a filter change is executed, and the fields are observed in another filter. These blocks have a minimum size that depends on the given exposure time and ensures that there are at least 30 minutes of observations between field reobservations. MUSHROOMS calculates the expected block length using an average slew time assumption of 10 s since the order of observations, which determines the actual slew time for each block, is not found until the next step. We use this block-division heuristic rather than giving the scheduler complete freedom to order fields and filter changes as it sees fit due to the computational complexity of complete freedom, which would require orders of magnitude more to run.

To minimize slew time within each block, MUSHROOMS calculates the slew times from each field within a block to all other fields in that block using a traveling salesperson (TSP) algorithm to find the order of observations that will minimize slew time within each block.

Finally, MUSHROOMS postprocesses the schedule to ensure that it is valid and satisfies all the requirements laid out in Appendix A.2. Because the block-division algorithm uses a fixed slew time, if one of the blocks has an average slew time higher than that, the block will run longer than expected, and all subsequent blocks will have to be delayed to avoid scheduling two observations at the same time. There is an edge case where this delay means MUSHROOMS schedules a field for observation when it is (barely) below the visibility requirement and cannot be observed. In this scenario, we automatically rerun the schedule utilizing a gap parameter to add more time between the offending blocks. However, in the 951 simulations utilized for this paper (see Section 4), it did not occur once.

The variables, constraints, and objective function behind MUSHROOMS are laid out explicitly in the Appendix. This algorithm is a modification of the classic max-weighted coverage problem (Nemhauser et al. 1978), with additional constraints to allow for the creation of valid observing blocks, as well as an additional (optional) penalty factor p in the objective function.

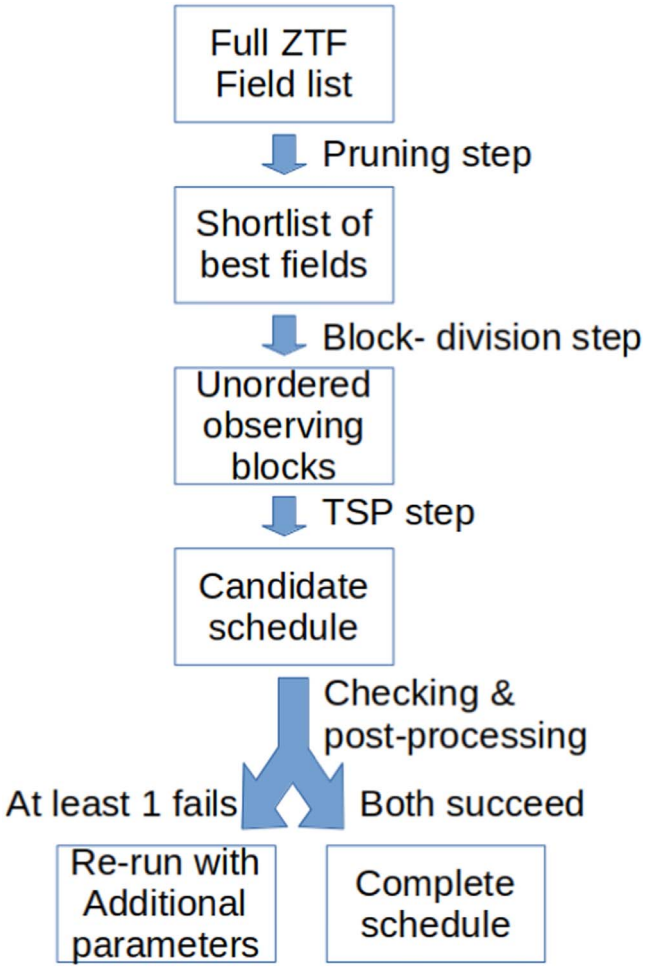

Figure 1. An illustration of MUSHROOMS' algorithm. TSP stands for traveling salesperson.

4. Simulated Observing Plans

To assess the efficiency of the generated schedules, we ran both the MUSHROOMS and `gwemopt` algorithm on 951 simulated binary neutron star detections consistent with the third LIGO–Virgo observing run (O3) from Petrov et al. (2022). Both `gwemopt` and MUSHROOMS were used to schedule follow-up observing plans for the 24 hr immediately following each simulated detection. Without the time to fine-tune p for each skymap, MUSHROOMS was run with $p = 0$ for all skymaps. An example schedule can be seen in Figure 2.

When performing the block-division algorithm, we recorded the runtime of all 951 schedules, which can be seen in Figure 3. The mean and median runtimes for this step were 107 and 15 s, respectively. The large difference between the mean and median can be attributed to the 140 schedules that took the entire time limit of 500 s to complete. In these cases, the solver would quickly converge on a high-quality solution, but would then spend the rest of the time limit attempting to narrow the optimality gap. This 500 s time limit was chosen because when developing MUSHROOMS it was observed that most schedules that converged in a reasonable amount of time did

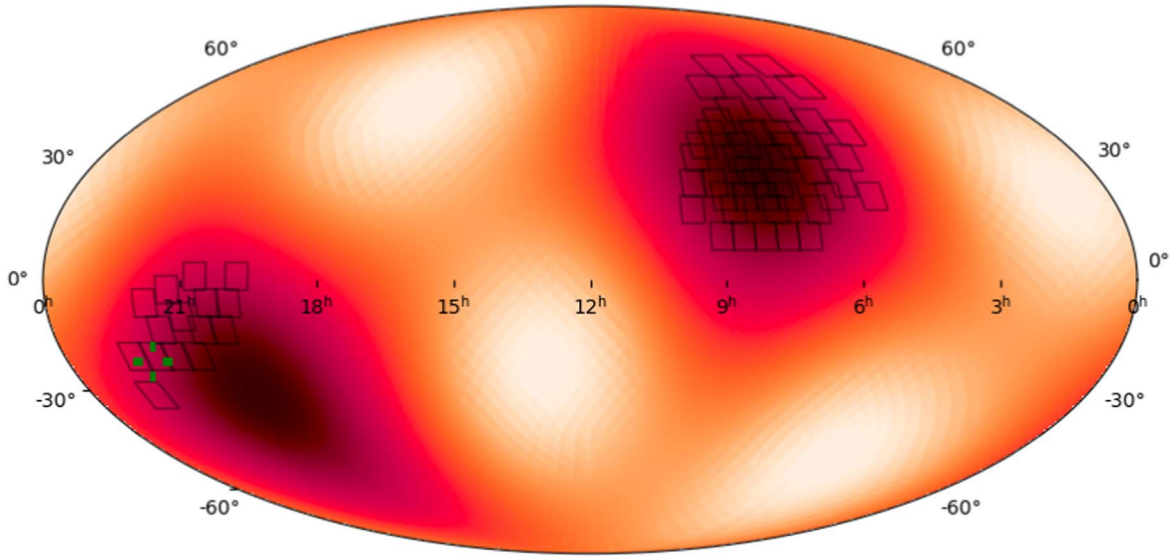

Figure 2. Skymap of simulation 119 from Petrov et al. (2022)and the schedule created for it by MUSHROOMS. The fields selected for observation are outlined in black, and the true location of the kilonova event is highlighted with a green cross.

so before 500 s, though it could easily be lowered to even 100 s without a substantial decrease in efficiency. Only an additional 64 schedules would be truncated and use the near-optimal candidate solutions instead of a solution proven to be globally optimal. Additionally, with a maximum number of six blocks (and thus a maximum of six filter changes in a night), the average number of filter changes scheduled was 4.7.

4.1. Comparisons to `gwemopt`

For all 951 skymaps we first measured the total probability density observed by each schedule, hereafter referred to as "probability coverage." MUSHROOMS saw an average probability coverage of 0.418, while `gwemopt` had an average probability coverage of 0.387, an 8.0% increase in probability coverage; however, MUSHROOMS' schedules had an average runtime of 23,700 s, while `gwemopt` had an average runtime of 16,800 s, a 41.5% increase in runtime. This is because MUSHROOMS was run with $p = 0$, meaning it filled all available time, while `gwemopt` has some logic to stop when it gets diminishing returns by adding more fields to observe.

To make a more equal comparison, we focused on the skymaps where MUSHROOMS and `gwemopt` made no more than six additional observations compared to the other. This value was chosen because it kept the difference in the average runtime of the schedules low, while still including a large amount of skymaps. The average runtimes were 22,900 s for MUSHROOMS and 22,800 s for `gwemopt` over this subset of 329 simulated events. An important note here is that there is selection bias here toward skymaps with a greater 90% credible area, since they are the ones that `gwemopt` usually makes longer schedules for, as well as more well-localized schedules that, due to the event time and location, MUSHROOMS and `gwemopt` both filled almost all available time. A frequency histogram comparing the area distributions can be seen in Figure 4.

For this subset, MUSHROOMS has an average probability coverage of 0.353, while `gwemopt` has an average probability coverage of 0.333, only a 5.8% improvement for MUSHROOMS. For a skymap-by-skymap comparison, Figure 5 is a scatterplot comparing the probability coverages achieved by MUSHROOMS and `gwemopt` over this subset.

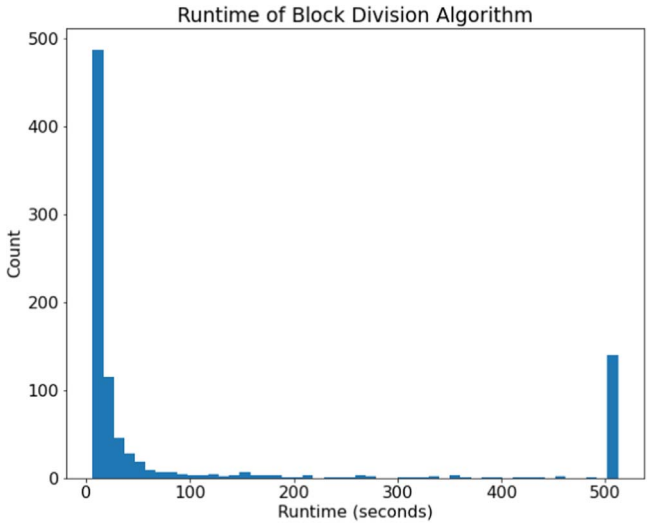

Figure 3. Runtime of the block-division algorithm. The large number of schedules clustered at 500 s is a result of setting a 500 s time limit for this step of optimization; all 500 s runtimes are when the MILP solver used would converge quickly on a high-quality solution but then spend the duration trying to lower the optimality gap.

An important note, however, is that MUSHROOMS does not always outperform `gwemopt`, even if the solution was not truncated by the 500 s time limit. This is because, even though the solution found is an optimal solution for MUSHROOMS' block-division heuristic, it may not be a globally optimal schedule. The design of MUSHROOMS forces the solutions to take on a certain format with observing blocks that are repeated in two different filters. This means the problem is (comparably) easy to implement using MILP and runs quickly, but if the best possible schedule does not fit such a format, MUSHROOMS cannot produce it. `gwemopt` has more freedom in ordering filter changes and block observations, meaning it can sometimes surpass MUSHROOMS, even with a greedy algorithm. Producing a more complex MILP formulation that lacks these

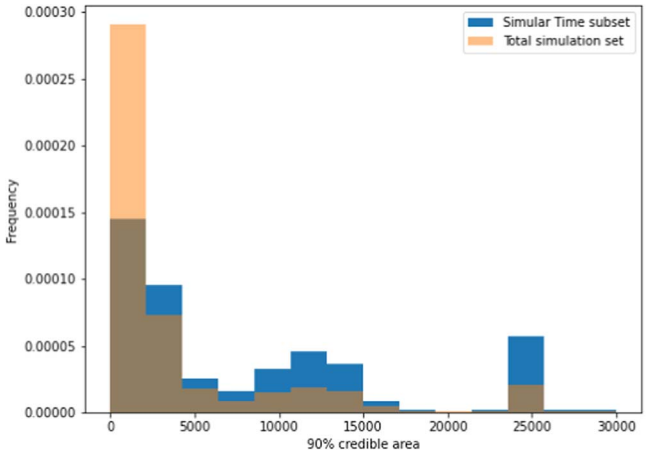

Figure 4. Frequency histograms of 90% credible area for all 951 simulations and the 329 where MUSHROOMS and `gwemopt` produce schedules of a similar length.

restrictions and can always surpass `gwemopt` is an avenue for future research.

For now, this means we can use a hybrid scheduler to achieve better results than either MUSHROOMS or `gwemopt` alone. By running both MUSHROOMS and `gwemopt` and using the schedule with a higher probability coverage, we can get an average coverage of 0.360, an 8.1% improvement over `gwemopt` alone and a 2.1% improvement over just MUSHROOMS.

4.2. Detection Efficiency Characterizations

Probability coverage, however, is not equivalent to the actual performance a schedule will have, since it fails to capture the difficulties in identifying a kilonova; even if the field containing it is observed, a kilonova might not be detected due to it being too dim to significantly differ from the reference. As fast-fading transients, kilonovae vary in magnitude significantly even over the 24 hr both schedules were allotted to search, meaning that the order of observations has a significant impact on the schedule's quality not captured by probability coverage. To address this, following Petrov et al. (2022), we characterized the resulting schedules' efficiencies with `gwemopt`'s simulation and injection recovery suite for two different kilonova light-curve models. This is done by injecting 10,000 kilonovae into the sky following the skymap's probability distribution, and each schedule's efficiency is the proportion of those kilonova that the ZTF would have been able to detect following each schedule. The light-curve models for the kilonovae used here, an optimistic and a conservative model, were generated by the radiative transfer code POSSIS (Bulla 2019) and summarized in Dietrich et al. (2020b). For details about the physical properties of each light curve, see Table 1.

Tables 2 and 3 compare the efficiencies of MUSHROOMS, `gwemopt`, and the hybrid implementation of the two, for all skymaps (Table 2), and for just the ones where both schedules are of a similar length (Table 3). Due to the large number of simulations, the Monte Carlo uncertainty in these values is negligible. In both cases, the hybrid method outperforms both schedulers acting on their own, with an efficiency increase of about 11.5% (11.1%) for a conservative (optimistic) light curve

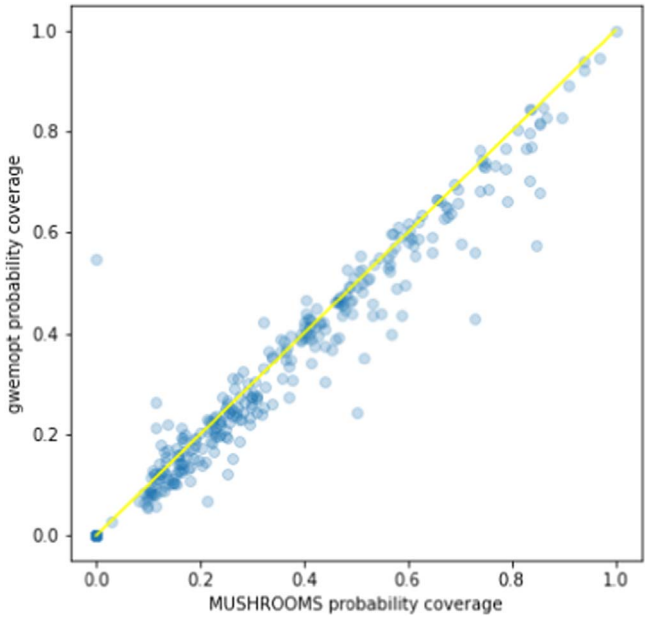

Figure 5. Probability coverage achieved by both schedulers over the similar schedule-length subset. The line $y = x$ is added in yellow for ease of comparison, with dots below the line representing schedules where MUSHROOMS outperforms `gwemopt`.

Table 1
Kilonova Light-curve Model Parameters

	Optimistic	Conservative
Dynamical ejecta mass ($M_\odot$)	0.005	0.01
Wind ejecta mass ($M_\odot$)	0.11	0.01
Half opening angle	45°	45°
Peak g-band absolute magnitude	−15.7	−15.1
Peak r-band absolute magnitude	−16.0	−15.7

Table 2
Scheduler Efficiencies for All Skymaps

	Optimistic	Conservative
MUSHROOMS	0.22	0.21
`gwemopt`	0.21	0.20
Hybrid	0.25	0.23

Table 3
Scheduler Efficiencies for Similar-length Subset

	Optimistic	Conservative
MUSHROOMS	0.186	0.163
`gwemopt`	0.180	0.156
Hybrid	0.200	0.174

in the subset where both schedules are of the same length compared to just using `gwemopt` alone.

Figure 6 compares the 90% credible area of each skymap to the percent improvement in efficiency that would result from utilizing the hybrid method as opposed to `gwemopt` to produce a schedule for it. The selection bias against more well-localized skymaps is clear, as well as MUSHROOMS'

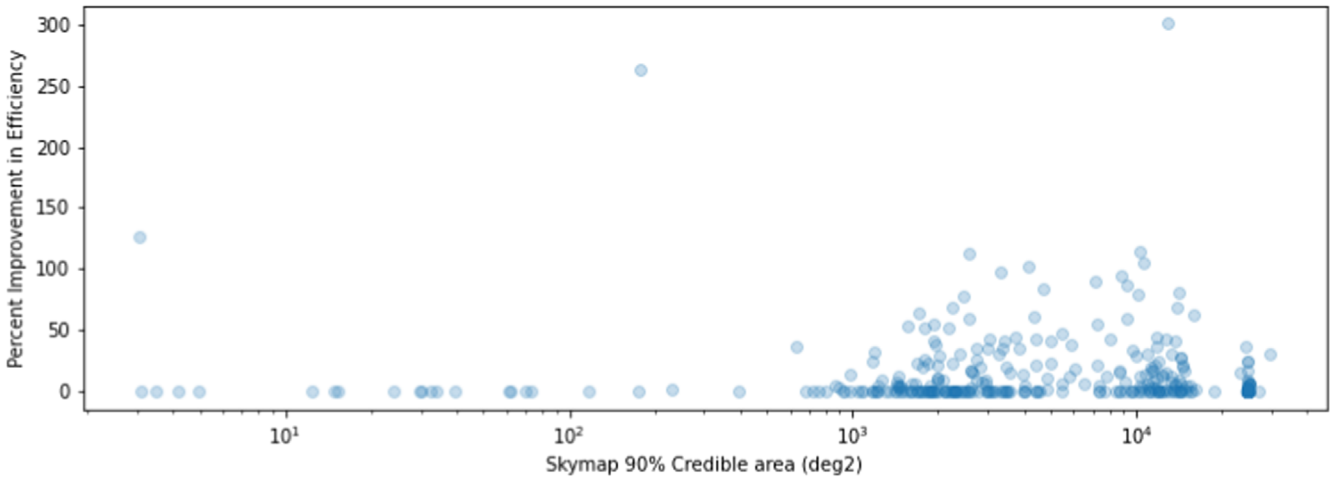

Figure 6. Percent improvement in detection efficiency by utilizing the hybrid scheduling method. A 0% improvement means that `gwemopt` performed better than or equal to MUSHROOMS for that skymap. The three outliers at above 100% improvement are schedules that had low efficiencies when run with `gwemopt` and a small absolute efficiency increase from MUSHROOMS resulted in a large relative increase.

comparative weakness at scheduling for these more localized detections. Only 37 out of 97 (38.1%) of detections below 1000 deg^2 were improved upon by mushrooms, while 131 out of 232 (56.5%) of detections above 1000 deg^2 were improved upon by MUSHROOMS.

Because these smaller localizations where MUSHROOMS does worse make up a larger proportion of the total observations, this means that in actual use employing a hybrid MUSHROOMS-`gwemopt` strategy will result in less than a 11.5% (11.1%) improvement in detection efficiency for a conservative (optimistic) light curve. Additionally, since the hybrid strategy will never do worse than `gwemopt` alone, for a worst-case scenario, where MUSHROOMS is better for none of the remaining 622 skymaps when schedule lengths are equal, that will result in a minimum 3.1% (3.2%) efficiency increase for a conservative (optimistic) light curve, establishing an upper and lower bound of 11% and 3% respectively on the potential performance improvement of applying this hybrid method in real observing scenarios.

5. Conclusion and Outlook

In this paper, we presented a novel scheduling algorithm for scheduling a wide field-of-view survey follow-up for multi-messenger events, outlined its MILP formulation, and compared its performance to `gwemopt`, the target-of-opportunity scheduler used by the ZTF and other surveys in recent observing runs. We focused on the MUSHROOMS block-division algorithm, outlining the parameters, decision variables, objective function, and constraints used to define this problem. Fundamentally, the block-division algorithm is an alteration of a max-weighted coverage problem, but instead of simply choosing a certain number of fields to look at, the algorithm assigns fields to variable-length blocks that are under further constraints to ensure all fields within them are observable and that no two blocks overlap. We include an additional optional penalty factor introduced into the objective function that allows for one to only observe fields that add enough probability coverage to overcome the penalty factor, leading to shorter schedules that infringe less on other programs. We also introduce a postprocessing step to check for block overlaps that could be introduced by the fixed slew time approximation.

Next, we compared MUSHROOMS to `gwemopt`, with MUSHROOMS achieving similar efficiencies to `gwemopt` for both light-curve models used. We showed that when the differing strengths between MUSHROOMS and `gwemopt` mean are used in concert, one is able to achieve efficiencies 3% to 11% higher than `gwemopt` alone.

The algorithm behind MUSHROOMS is a comparatively straightforward one, designed to quickly run on everyday computer hardware while still producing efficient schedules, and it already is able to increase the detection efficiency when used alongside the previous greedy scheduler. This shows not only the potential of using mixed-integer linear programming for scheduling a multimessenger target-of-opportunity follow-up, and for observational scheduling as a whole, but also that there is significant room for improvement in MUSHROOMS or another mixed-integer scheduler, since problem formulation's rigidity in its schedules means it can still be outdone by `gwemopt` for some schedules.

There are a number of planned improvements for MILP schedulers for multimessenger follow-up. For example, MUSHROOMS does not account for the moon distance and lunar phase when scheduling observations. MUSHROOMS also does not have a straightforward way to respond to weather and other unexpected events. Currently, one would have to edit the input skymap, setting probabilities associated with affected HEALPix to zero before renormalizing and inputting it to MUSHROOMS. Improving it to account for both of those is important future work. Potentially more important, it treats the source as having constant flux for the duration of the schedule, which is not correct for the fast transient kilonova models considered here from Dietrich et al. (2020b). The most straightforward way to address this issue would be to accept a desired light-curve model as an additional parameter and to make an alteration to the objective function of the block-division step, using that model to alter the weight of each pixel by when it is observed, such as multiplying the weight associated with that pixel by the ratio of the light-curve magnitude at the observation time to the maximum magnitude. As the complete field order is not determined until the traveling salesman problem step, one may have to use an approximation of when each pixel is observed, such as the midpoint of the first block to observe a given pixel.

The block-division formulation, while a useful heuristic for limited time and computing power, has some limitations, especially when variable exposure times are desired. Producing a model that is not constrained by blocks and can jointly be optimized over the selection and ordering of all fields, subject only to the observing and time constraints, would lead to more efficient schedules. Also, allowing the model to vary the exposure times of individual observations would lead to higher chances of detection because it would adjust for time and a position-dependent sky background. However, both improvements are much more computationally complex, and will require much greater optimization and application of high-level operation research techniques. Using the experience gained from working on this project, among others, several authors of this paper have begun development on a more general multifacility observation scheduling toolkit, which will add those considerations into its problem formulation.

We thank Alexander Criswell for their feedback when writing the abstract. B.P. acknowledges support from a Northeastern Lawrence Co-op Fellowship. M.W.C. acknowledges support from the National Science Foundation with grant Nos. PHY-2010970 and OAC-2117997. S.A. acknowledges support from the GROWTH National Science Foundation PIRE grant 1545949.

Software: `astropy` (Astropy Collaboration et al. 2013); `matplotlib`; `ligo.skymap`[8].

Appendix
MILP Formalism

The following is the variables, objective and constraints used to mathematically define the MUSHROOMS algorithm.

A.1. Parameters

$N_{\rm field}$ Number of fields received from pruning step (A1)

$N_{\rm pix}$ Number of HEALPix pixels (A2)

$\{S_l\}_{l=0}^{N_{pix}-1}$ Set of fields that contain pixel l (A3)

$\{w_l\}_{l=0}^{N_{pix}-1}$ The probability associated with HEALPix pixel l (A4)

$\{T_{s,j}, T_{e,j}\}_{j=0}^{N_{field}-1}$ The start and end observability times for field j (A5)

$b_{\rm max}$ The maximum number of observation blocks to schedule (A6)

$b_{\rm size}$ The minimum number of fields in an observation block (A7)

$t_{\rm exp}$, $t_{\rm slew}$, $t_{\rm filt}$ The exposure, slew and filter change times (A8)

$t_{\rm start}$, $t_{\rm end}$ Start and end times of the observing run (A9)

p Penalty factor for number of fields observed (A10)

$\{g_i\}_{i=0}^{i=b_{\rm max}-2}$ Additional time gap to be added between blocks i and i+1 (A11)

An important thing to note is that as the specific order of fields is not yet determined, t_{slew} is a fixed slew time approximation. The specific slew times between each field is determined in the TSP step.

A.2. Binary Decision Variables

$\{B_{i,j}\}_{i=0,j=0}^{b_{\rm max}-1,N_{field}-1}$ Is field$j$ observed in block i? (A12)

$\{y_l\}_{l=0}^{N_{pix}-1}$ Is HEALPix pixel l observed? (A13)

$\{U_i\}_{i=0}^{b_{\rm max}-1}$ Is block i used to make observations? (A14)

A.3. Continuous Decision Variables

$\{t_{o,i}\}_{i=0}^{b_{\rm max}-1}$ The starting time of block$i$ (A15)

A.4. Objective and Constraints

Maximize $\sum_l w_l y_l - p\sum_{i,j} B$ subject to the following constraints:

$$\forall i, \quad \sum_j B_{i,j} \geqslant b_{size} U_i \quad \text{Set minimum block size} \tag{A16}$$

$$\forall i, j, \quad U_i \geqslant B_{i,j} \quad \text{If a block makes at least 1 observation, it is being used} \tag{A17}$$

$$\forall i > 0, \quad U_i \leqslant U_{i-1} \quad \text{All unused blocks are at the end of the night} \tag{A18}$$

$$\forall l, \quad \sum_{i \in S_{l,j}} B_{i,j} \geqslant y_l$$

A pixel is observed if it is in any observed field (A19)

$$\forall i, j \quad t_{o,i} + 2 \cdot (t_{\rm exp} + t_{\rm slew}) \sum_{j'} B_{i,j'} + t_{filt} \leqslant B_{i,j}[T_{e,j} - t_{\rm start} - t_{\rm exp}] + (1 - B_{i,j})[t_{\rm end} - t_{\rm start}] \tag{A20}$$

A block's end time must be before the observability end time of all fields within it (A21)

$$\forall i, j \quad t_{o,i} \geqslant B_{i,j}[T_{s,j} - t_{\rm start}] \tag{A22}$$

A block's start time must be after the observability start time of all fields within it (A23)

$$\forall i > 0, \quad t_{o,i} \geqslant t_{o,i-1} + 2 \cdot (t_{\rm exp} + t_{\rm slew}) \sum_j B_{i,j} + t_{\rm filt} + g_{i-1} \tag{A24}$$

A block's start time must be after the previous block finishes (A25)

[8] lscsoft.docs.ligo.org/ligo.skymap

ORCID iDs

B. Parazin https://orcid.org/0000-0002-3155-0385
Michael W. Coughlin https://orcid.org/0000-0002-8262-2924
Leo P. Singer https://orcid.org/0000-0001-9898-5597
Vaidehi Gupta https://orcid.org/0000-0002-7672-0480
Shreya Anand https://orcid.org/0000-0003-3768-7515

References

Abbott, B. P., Abbott, R., Abbott, T. D., et al. 2017b, Natur, 551, 85
Abbott, B. P., Abbott, R., Abbott, T. D., et al. 2017a, PhRvL, 119, 161101
Abbott, B. P., Abbott, R., Abbott, T. D., et al. 2018, PhRvL, 121, 161101
Abbott, R., Abbott, T. D., Abraham, S., et al. 2021, PhRvX, 11, 021053
Ahumada, T., Singer, L. P., Anand, S., et al. 2021, NatAs, 5, 917

Almualla, M., Coughlin, M. W., Anand, S., et al. 2020, MNRAS, 495, 4366
Anand, S., Coughlin, M. W., Kasliwal, M. M., et al. 2020, NatAs, 5, 46
Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., et al. 2013, A&A, 558, A33
Bellm, E., Ford, E. B., Tohuvavohu, A., et al. 2019a, BAAS, 51, 125
Bellm, E. C., Kulkarni, S. R., Barlow, T., et al. 2019b, PASP, 131, 068003
Bellm, E. C., Kulkarni, S. R., Graham, M. J., et al. 2019c, PASP, 131, 018002
Bulla, M. 2019, MNRAS, 489, 5037
Chornock, R., Berger, E., Kasen, D., et al. 2017, ApJL, 848, L19
Coughlin, M. W., Ahumada, T., Anand, S., et al. 2019a, ApJL, 885, L19
Coughlin, M. W., Ahumada, T., Cenko, S. B., et al. 2019b, PASP, 131, 048001
Coughlin, M. W., Antier, S., Corre, D., et al. 2019c, MNRAS, 489, 5775
Coughlin, M. W., Dietrich, T., Antier, S., et al. 2020, MNRAS, 492, 863
Coughlin, M. W., Dietrich, T., Antier, S., et al. 2020, MNRAS, 497, 1181
Coughlin, M. W., Dietrich, T., Doctor, Z., et al. 2018, MNRAS, 480, 3871
Coughlin, M. W., Dietrich, T., Margalit, B., & Metzger, B. D. 2019, MNRAS, 489, L91
Coughlin, M. W., Tao, D., Chan, M. L., et al. 2018, MNRAS, 478, 692
Coulter, D. A., Foley, R. J., Kilpatrick, C. D., et al. 2017, Sci, 358, 1556
Cowperthwaite, P. S., Berger, E., Villar, V. A., et al. 2017, ApJL, 848, L17
Dekany, R., Smith, R. M., Riddle, R., et al. 2020, PASP, 132, 038001
Dietrich, T., Coughlin, M. W., Pang, P. T. H., et al. 2020a, Sci, 370, 1450
Dietrich, T., Coughlin, M. W., Pang, P. T. H., et al. 2020b, Sci, 370, 1450
Evans, P. A., Cenko, S. B., Kennea, J. A., et al. 2017, Sci, 358, 1565
Goldstein, A., Veres, P., Burns, E., et al. 2017, ApJL, 848, L14
Graham, M. J., Kulkarni, S. R., Bellm, E. C., et al. 2019, PASP, 131, 078001
Hallinan, G., Corsi, A., Mooley, K. P., et al. 2017, Sci, 358, 1579
Hotokezaka, K., Nakar, E., Gottlieb, O., et al. 2019, NatAs, 3, 940
Kasliwal, M. M., Anand, S., Ahumada, T., et al. 2020, ApJ, 905, 145
Kasliwal, M. M., Nakar, E., Singer, L. P., et al. 2017, Sci, 358, 1559
Kilpatrick, C. D., Foley, R. J., Kasen, D., et al. 2017, Sci, 358, 1583
Lampoudi, S., Saunders, E., & Eastman, J. 2015, arxiv:1503.07170
Masci, F. J., Laher, R. R., Rusholme, B., et al. 2018, PASP, 131, 018003
Morris, B. M., Tollerud, E., Sipocz, B., et al. 2018, AJ, 155, 128
Nemhauser, G. L., Wolsey, L. A., & Fisher, M. L. 1978, MatPr, 14, 265
Parazin, B. 2022, bparazin/MUSHROOMS: MUSHROOMS v1.0.0, Zenodo, doi: 10.5281/zenodo.6659827
Petrov, P., Singer, L. P., Coughlin, M. W., et al. 2022, ApJ, 924, 54
Pian, E., D'Avanzo, P., Benetti, S., et al. 2017, Natur, 551, 67
Radice, D., Perego, A., Zappa, F., & Bernuzzi, S. 2018, ApJL, 852, L29
Shappee, B. J., Simon, J. D., Drout, M. R., et al. 2017, Sci, 358, 1574
Singer, L. P., & Price, L. R. 2016, PhRvD, 93, 024013
Smartt, S. J., Chen, T. W., Jerkstrand, A., et al. 2017, Natur, 551, 75
Solar, M., Michelon, P., Avarias, J., & Garces, M. 2016, A&C, 15, 90
Troja, E., Piro, L., van Eerten, H., et al. 2017, Natur, 551, 71